METHOD AND SYSTEM FOR CREATING, SERVING AND TRACKING ADVERTISEMENTS

[001] A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

CROSS REFERENCE TO RELATED APPLICATIONS

[002] The present applications claims the benefit of US Provisional patent Application Serial No. 60/421,762, entitled METHOD AND SYSTEM FOR CREATING, SERVING AND TRACKING ADVERTISEMENTS, filed October 28, 2002, which application is hereby incorporated herein by reference.

BACKGROUND OF THE INVENTION

1.   Field of the Invention

[003] The present invention relates generally to providing advertisements (ads) via a network, such as the Internet and, more particularly, to methods and systems for providing creation, serving and tracking of ads.

2.   Description of Related Art

[004] With the advent of the Internet, an industry has arisen focused on serving ads. Such industry includes marketers or advertisers, which typically undertake the marketing of their own products and services, ad agencies, which typically undertake planning, creating, executing and tracking ad campaigns on behalf of their clients, and web publishers, which typically operate one or more websites and sell ad space on such sites. The tracking of ads and ad campaigns

generally involves monitoring the effectiveness of each ad, for example, by the number of times the ad is viewed by a users (often referred to as a "page view"), the number of times a user activates a link in the ad to the advertiser's web page (often referred to as a "click-through"), the number of times a user makes a purchase after activating a link in the ad (often referred to as an "acquisition"). To effectively track ads, the computer code that comprises the ad contains code that monitors the relevant metrics.

[005]   Numerous companies have been created to provide software tools to assist such entities involved in the advertising industry. These tools often purport to simplify and increase the efficiency of the planning, execution and analysis of ad campaigns. Despite such tools, ads and ad campaigns can be difficult to create, execute and analyze.

[006]   For example, the processes of providing, or "serving," the ads and tracking the ads can be technically complicated, especially where one entity is serving a large number of ads. An ad is typically an item of computer code that is pulled from an ad server and incorporated into the desired Web page. In general, the entity serving the ads receives the advertising content from the advertisers or agencies and must incorporate the content into the ad code. Such a process is labor intensive and requires the ad-serving entity to maintain a staff of programmers. These programmers must be used not only to incorporate the received ad content into the ad code to be served, but also to effectuate changes to the ad content received from the advertiser or agency. Such work is labor intensive and results in both decreased profits for the entities serving the ads and, to the extent the cost is not completely absorbed by the entity serving the ad, increased advertising costs to the advertiser and/or agency.

[007]   To make matters worse, there are no meaningful economies of scale. Working

with more ad agencies and serving more ads only requires a greater number of programmers to

address these problems. In short, serving more ads exacerbates the problems.

[008]   The existing ad tools do not adequately address these problems. Accordingly,

there exists a need for an improved system and method for creating, serving and tracking ads.

SUMMARY OF THE INVENTION

[009]   These and other needs are satisfied by the present invention. According to one

embodiment, a method for providing an ad is disclosed. The method comprises combining an ad

input file with a conduit file to create an integrated ad file, wherein the ad input file identifies at

least the content of the ad and the conduit file identifies at least the tracking data for the ad. The

integrated ad file may then be served. According to a further embodiment, the method further

includes effectuating a modification to the ad content by updating the ad input file, recombining

the modified ad input file with the conduit file to create a modified integrate ad file and serving

the modified integrated ad file.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010]   The following drawings, which are attached hereto and form a part hereof, are

exemplary in nature and are not meant to be limiting of the scope of the present invention.

[0011]   Figure 1 is an overall schematic illustrating the system according to one

embodiment of the present invention.

[0012]   Figure 2 is a schematic illustrating the process flow of the system according to

one embodiment the present invention.

[0013]   Figure 3 is html code for calling an ad, according to one embodiment of the

present invention.

[0014]   Figure 4 is a JavaScript file called by the html code of Figure 2, according to one embodiment of the present invention.

[0015]   Figure 5 is a schematic illustrating the process flow at the portal, according to one embodiment of the present invention.

[0016]   Figure 6 is a screen shot of a portion of an ad input file, according to one embodiment of the present invention.

[0017]   Figure 7 is a screen shot of a portion of a conduit file, according to one embodiment of the present invention.

[0018]   Figure 8 is a screen shot of a portion of an integrated ad file, according to one embodiment of the present invention.

[0019]   Figure 9 is a screen shot of another portion of an integrated ad file, according to one embodiment of the present invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0020]   Certain embodiments of the present invention will now be described in connection with the aforementioned. Turning first to Figure 1, an exemplary system according to one embodiment is shown. As illustrated, portal 100 includes several components for serving and tracking ads. In this regard, portal 100 includes one or more ad servers 102 for serving ads, one or more tracking servers 104 for tracking user responses to ads, one or more web servers 106 for providing web pages and one or more workstations 108 in communication via a network, such as a local area network (LAN). The LAN, in turn, is coupled to a network, such as the Internet. Although not separately shown, portal 100 also includes data storage for providing the functionality described herein. Furthermore, it is to be understood that the particular type and

arrangement of components is not essential to the present invention; for example, a single server could provide the function of both the ad server and tracking server.

[0021]   Although the portal 100 is described as serving and tracking the ads, it is to be understood that it is within the scope of the present invention to have one or more different entities serving and/or tracking ads. Furthermore, although the term "portal" is used to describe the entity, the term is meant to encompass all types of entities, includes those operating web portals, such as Yahoo! Inc., advertisement technology companies, such as DoubleClick, Inc., e-commerce sites, such as Cnet.com, and the like. Similarly, reference to advertiser 110 is meant to encompass any entity requesting an ad be served or tracked, including the advertiser, ad agency, and other entities involved in the ad process.

[0022]   The portal 100 is in communication, either electronically or otherwise, with an advertiser 110. In general, the advertiser, on behalf of itself or its clients, creates ads to be served to users 112 via the Internet. As described in greater detail below, the advertiser 110 submits an electronic ad input file to the portal 100 that contains the ad message, specifying the required text, graphics and other content in the ad. The ad may be part of a webpage served by the portal's web server 114 or by another web server 114, for example, one operated by the advertiser 110. The ad may also include one or more buttons (or other actions) that allow users 112 to link to other websites, for example, those provided by the web servers 114 of the advertiser or its clients.

[0023]   Notably, the electronic file submitted by the advertiser 110 in the present embodiment, is in certain respects, of a standard format, regardless of the particular ad content and tracking methodology being use. This standardized format provides greater ease and flexibility in editing and serving the ad and tracking the responses of users 112. More

specifically, because the ad content is merged with the code necessary for tracking the ad, as described below, the advertiser 110 may provide the content to many different portals without concerning itself with the difference in portal tracking methodologies. Similarly, portals may select a tracking methodology and implement it independently from the ad content. On one hand, advertisers may modify the ad content without implicating changes to the tracking code, and on the other hand, portals may modify the tracking methodology and code without implicating changes to the ad content.

[0024]   The users 112 access the Internet with any suitable hardware and software, for example, a personal computer running an Internet browser, such as that provided by Microsoft Corporation under the trade name INTERNET EXPLORER. In the present embodiment, the ad is prepared using Macromedia Flash, for example, version 5. Therefore, the user 112 also has Flash software plug-ins necessary to view the ads.

[0025]   Having identified the entities related to the present embodiment, operation of the present embodiment will now be described in greater detail with reference to Figures 2-9.

[0026]   Figure 2 illustrates the general operation of the present embodiment in which a user 112 accesses a web page, for example, the portal's home page provided by web server 106 or another page provided by a web server 114. The web page includes html code (ad.html) 202 for loading an integrated ad file 204 from the ad server 102. As will be described in greater detail below, the integrated ad file 204 reflects the integration or combination of two separate files: an ad input file, primarily including the contents of the ad, and a conduit file, primarily including code used in tracking the ad. Because Macromedia Flash is used in the present embodiment, the files have ".swf" extensions; thus, the integrated ad file is generically referred to as "ad_done.swf"; the electronic ad input file 504 submitted by the Advertiser 110 is

generically referred to as "ad_input.swf"; and the conduit file created by the portal 100 is

generically referred to as "conduit.swf."

[0027]   The html code 202 also calls a JavaScript file ("sniffer.js") 206. As will be

described in greater detail below, the html code 202 declares a query string variable for the ad,

and the integrated ad file 204 uses the declared query string values from the html code 202 to

perform functions specified within the integrated ad file 204. Additionally, when a user 112

activates a button in the ad and is linked to another web page, the integrated ad file 204 acquires

relevant tracking data and provides it to tracking server 104.

[0028]   The html code 202 and JavaScript 206 will be described in greater detail below

with reference to Figures 3 and 4, respectively.

[0029]   Exemplary html code 202 will now be described with reference to Figure 3. In

general, the code 202 relates to several functions, including: (1) indicating to the ad server 102

whether the user 112 activating a button to link to another web page should open that web page

in the same or different window; (2) providing an array of uniform resource locators (URLs),

which correspond to variables in the integrated ad file 204; and (3) declaring the query string and

other variables related to tracking the ad by the integrated ad file 204.

[0030]   Determining whether or not a user's link to another web page opens in a same or

different window involves the integrated ad file 204 pulling the "targID" variable specified in the

html code 202. As described in greater detail below with regard to Figures 5 and 7, the

integrated ad file 204 includes code (incorporated from the conduit file also described below)

that pulls the value of the "targID" variable and, depending on the value of the variable, either

provides the ad in a new window or in the same window.

[0031]   The html code 202 also specifies the URLs associated with the buttons of the ad. To accomplish this, the code 202 includes an array of URLs (at lines 4 - 5). In the present embodiment, these variables are "URL1" and "URL2." As will be appreciated by those skilled in the art, the invention is scalable in that greater or fewer URLs may be specified in the ad and html code. As will be discussed in greater detail below in connection with Figures 5 and 6, the URLs specified in the html code 202 are pulled by the integrated ad file 204 and associated with the button actions in the ad.

[0032]   The html code 202 also includes several lines of code and variables directed to tracking the ad. In this regard, the code 202 includes a unique tracking identifier (var track) and, in the query string, a tracking string (adID='track+'). The unique tracking identifier specifies a redirection URL and a tracking variable. As is known in the art, the redirection URL is used to provide tracking information to the tracking server 104. When the ad is served, the integrated ad file 204 appends the appropriate tracking data to the redirection URL. In the present embodiment, such tracking data includes the relevant invoice order for the ad campaign, the line number for the ad campaign, the placement of the ad on the web page, the content identifier (ID) for the ad, which identifies the ad in the content management system of the portal 100, and the ad ID, which is a unique identifier for the ad. It is to be understood, however, that the particular tracking data utilized is not essential to the present invention and that the present invention encompasses tracking more or less data, including, for example, non-click-through data, such as changing the volume of an ad, pausing an ad, loading a different ad, and the like.

[0033]   Because the html code 202 loads the integrated ad file 204, the code 202 also includes the server and path where the integrated ad file resides ("var flashfile"). Similarly, the code 202 also specifies the path of additional ad files used in the ad ("var path"; "ap='+path+'").

-8-

The path information is included for such additional ad files, e.g., where the integrated ad file 204 loads additional movies, such as sounds associated with a button other than an exit event.

[0034]   The html code 202 also specifies the width ("var width") and height ("var height") of the ad, as well as the URL for an alternate image ("var altURL") and the path for the alternate image ("var altimg").

[0035]   Figure 4 illustrates the JavaScript file 206 called by the html code 202. In general, the JavaScript file 206 functions to determine whether the user 112 has at its computer the appropriate browser and Flash plug-ins. If the JavaScript file 206 determines that the plug-ins do not exist at the user 112, then the file 206 supplies the alternative .GIF, .JPG or other formatted image specified in the html code 202.

[0036]   Figure 5 illustrates operations performed by the portal 100 (e.g., an administrator 108) in creating the integrated ad file 204 to be served via the ad server 102. As illustrated, the portal 100 uses a merge tool 502 to combine the contents of an ad input file 504 provided by the Advertiser 110 with the contents of a conduit file 506 created by the portal 100. In general, the ad input file 504 contains the content of the ad, as provided by the advertiser, and the conduit file 506 contains code for tracking the ad.

[0037]   The merge tool 502 of the present embodiment is an executable program that combines the content of the ad file 504 with the contents of the conduit file 506 to create an integrated ad file 204 ("ad_done.swf"). The merge tool 502 may take any number of forms, may have any number of different user interfaces, and may be written in any number of languages, including C++, Java, Perl, and the like. As such, the merge tool make take the form of a web-based application accessible via an Internet Web page.

[0038]   In the present embodiment, the merge tool 502 replaces an empty movie clip object in the ad input file 504 with the conduit file 506, which is a one pixel by one pixel ad (and therefore does not appear in the ad) that includes additional lines of code directed to the tracking information. More specifically, the merge tool 202 of the current embodiment inserts the conduit file 506 into the ad input file 504 by first identifying in the ad input file 504 an empty movie clip into which the conduit file 506 will be inserted. As will be apparent to those skilled in the art, it is preferable to insert the conduit file 506 into an empty layer of the ad input file 504 so that the contents of the conduit file 506 do not obscure other aspects of the ad. Such movie clip can be thought of as a "placeholder" for the conduit file 506. To identify the empty movie clip, it is given a predefined name (e.g., "conduit_mc"). The advertiser 110 can be apprised of the need for an empty movie clip layer with the predefined name, which can be inserted in all ad input files when created. Once the merge tool 202 locates the empty layer by name, the merge tool 202 converts the conduit file 506 into the movie clip in the ad input file 504, and creates the new integrated ad file 204.

[0039]   In an alternate embodiment, rather than include an empty movie clip object, the ad file uses a load movie call to a new Flash layer, into which the conduit file 506 is merged. In general, any method that allows the contents of the files 504, 506 to be combined may be used. Furthermore, although the present embodiment involves incorporating the contents of the conduit file 506 into the ad input file 504, alternate embodiments involve incorporating the ad input file into the conduit file and combining the two files into a third file (e.g., a third file having two empty movie clip objects, one for each the ad input and conduit file; or a third file with two load movie calls; or a third with one empty movie clip object and one load movie call). The

integrated ad file 204 is, in essence, a restructured ad file that includes scripts for presenting and handling tracking of the ad.

[0040]   Those skilled in the art will appreciate that the present embodiment need only load a single file—the integrated ad file 204—when serving the ad. The integration of the ad input file 504 and the conduit file 506 prior to serving the ad permits this efficiency. An alternative is loading a first ad file (for example, one file containing the content of the ad), which, in turn, loads a second ad file (for example, one file containing the tracking information). Such alternative has the disadvantages of utilizing additional bandwidth of the ad server (i.e., loading of the single integrated ad file verses loading two separate files) and the potential for failure of the second ad file to load properly. Indeed, with a portal or other provider serving numerous ads, these disadvantages make the alternative of serving two separate files commercially unworkable.

[0041]   Exemplary ad files will now be described in greater detail with reference to Figures 6-9. Figure 6 illustrates a portion of the exemplary ad file 504 (frame 1), as viewed through an ActionScript viewer. In the present embodiment, the Advertiser 110 has created an ad with two buttons, each of which allows the user 112 viewing the ad to be linked to a different URL. These URLs are not included in the ad file 504 itself, but rather ad file 504 includes an exit code for each URL ("htmlURL1", "htmlURL2") that points to the URL provided in the HTML code 202 that calls the integrated ad file 204. More specifically, as noted above in connection with Figure 3, the html code 202 includes a query string that correlates the URL variable from the ad input file 504 to a URL in the URL array of the html code 202 ("htmlURL1='+URL!+'") (where URL1 is stated to be "http://www.yahoo.com" and URL2 can be, for example, the advertiser's website, represented by "http://www.advertiser.com").

-11-

[0042]   As will be appreciated by those skilled in the art, by including only a variable (exit code) in the ad input file 504 and including the actual URL in the HTML code, greater flexibility is provided to change the URLs. In other words, if the URLs were included in the ad file 504 and the ad file 504 was provided by the Advertiser 110, the Advertiser 110 would need to change the URLs and provide the portal 100 a new ad file 504 for each such change. In contrast, the present embodiment allows for changes to the URLs to be made to the html code 202 that calls the integrated ad file 204 without necessarily involving the entity serving the ad. Nevertheless, in alternate embodiments in which such flexibility is not required, the ad file 504 includes one or more URLs associated with one or more button actions.

[0043]   Figure 7 illustrates a portion of an exemplary conduit file 506, as viewed through an ActionScript viewer. More specifically, Figure 7 illustrates the ActionScript associated with frame 4 of the file 506. As will be appreciated by those skilled in the art, the conduit file 506 provides ActionScript for tracking the ad and for determining whether the ad opens in the same window in which the html code 202 is embedded or in a new window.

[0044]   In connection with tracking the ad, the ActionScript builds the redirection URL used in tracking. To this end, the conduit file 506 sets forth the root level of the integrated ad file 204 ("_level0:/"), which references the "adID" variable of the query string in the html code 202 that loads the integrated ad file 204 from the ad server 102.

[0045]   As will be apparent to those skilled in the art, the ActionScript from the conduit file 506, when merged into the integrated ad file 204, references variables that are populated by the html code 202 (e.g., "adID" corresponds to "+ track + adID" corresponds to "http://rd.yahoo.com/%space%").

[0046]   Furthermore, after the integrated ad file 204 is live on a web site (e.g., the portal's website or any other web site), the integrated ad file 204 uses the stated variable "track" and appends the exit code URL string (as defined in the html code 202) to the track URL string (Track url + Exit Url).

[0047]   To determine whether or not the ad should be opened in a new or the same window, the conduit file 506 includes an "if" statement to determine, based on the value of the "targID" variable in the html code 202, whether the ad opens in a new window or the same (i.e., parent) window. As illustrated, the "if" statement operates on the "targetURL" variable, which comprises the root level of the integrated ad file 204 and the "targID" variable populated from the html code 202. In the present embodiment, if the "targID" variable equals "_top", then the URL in the URL array of the html code 202 corresponding to the exit code in the integrated ad file 204 will open in a new window. If the value of the "targID" variable is "_blank" or something other than the "_top", then the URL opens in a new window. More specifically, this functionality is achieved by concatenating the redirection URL ("redirectURL") with a window open call, as set forth in the "if" statement ("wind"). Thus, the script "getURL(redirectURL,wind)" is used.

[0048]   It will be appreciated by those skilled in the art that including such functionality in the conduit file 506 provides the entity serving the ad with greater flexibility and control over the presentation of the ad. In short, the portal or other entity serving the ad can simply change the conduit file 506 locally to adjust the display of the ad, thereby retaining greater control over presentation of the ad. It is also within the scope of the present invention to include the functionality for determining whether to open a new window in any of the other files described

herein or even a separate file. In one such embodiment, the JavaScript file 206 includes code

that utilizes the open window function to determine whether to open a new window, as follows:

[0049]   function openWindow(x) {

[0050]   if (targID = = "_top") {

[0051]   top.location=URL[x]

[0052]   } else {

[0053]   window.open(URL[x]);

[0054]   }

[0055]   }

[0056]   In this alternate embodiment, the JavaScript file reads the value of the "targID"

variable in the html code 202. In the present embodiment, if the variable equals "_top", then the

URL corresponding to the "xth" URL in the URL array of the html code 202 will open in a new

window. If the value of the "targID" variable is something other than the "_top", then the

JavaScript function "window.open" causes the URL to open in a new window. As will be

appreciated by those skilled in the art, such an embodiment also provides advantages of

flexibility and ease of implementation.

[0057]   The conduit file 504 also includes an optional unique identifier. The unique

identifier is any special text that the entity serving the add can add (e.g., "R=1

Flash_ad_first_url_click") for making advertising statistics easier to read. More specifically,

using the identifier allows the ad to be identified with meaningful text, rather than with numbers.

[0058]   In certain embodiments, portal 100 includes additional code that acts as a filter.

For certain URLs the filter overrides or sets a default value for the targID value, thereby

dictating whether the ad opens in the same or different window. The filter may reside at the ad

server 102 or any other server associated with the portal 100. For example, in one embodiment, where the ad html code is part of an Internet e-mail page, an e-mail server includes the filter and causes the html code to specify that the ad open in the same window being provided to the user 112. Such filter ensures the user 112 remains at the e-mail page, rather than being redirected to another web page.

[0059]   Figure 8 illustrates a portion of the integrated ad file 204. As illustrated, frame 1 of the ad file 204 includes the references to the URLs contained within the ad input file 504. Similarly, Figure 9 illustrates that frame 4 of the integrated ad file 204 contains the ActionScript from the conduit file 506 for the same frame. As such, the merge tool 202 has replaced the empty movie clip object of the ad input file 504 with the contents of the conduit file 506.

[0060]   Those skilled in the art will recognize that the method and system of the present invention has many applications, may be implemented in many manners and, as such, is not to be limited by the foregoing exemplary embodiments and examples. In this regard, any number of the features of the different embodiments described herein may be combined into one single embodiment. Moreover, the scope of the present invention covers conventionally known and future developed variations and modifications to the system components (e.g., hardware, software, programming languages, communication networks, etc) described herein, as would be understood by those skilled in the art.